The Statistical Benefits of Quantile Temporal-Difference Learning for Value Estimation

Mark Rowland [1] **Yunhao Tang** [1] **Clare Lyle** [1] **Rémi Munos** [1] **Marc G. Bellemare** [2] **Will Dabney** [1]

Abstract

We study the problem of temporal-difference-based policy evaluation in reinforcement learning. In particular, we analyse the use of a distributional reinforcement learning algorithm, quantile temporal-difference learning (QTD), for this task. We reach the surprising conclusion that even if a practitioner has no interest in the return distribution beyond the mean, QTD (which learns predictions about the full distribution of returns) may offer performance superior to approaches such as classical TD learning, which predict only the mean return, even in the tabular setting.

1. Introduction

Distributional approaches to reinforcement learning (RL) aim to learn the full probability distribution over random returns an agent may encounter, rather than just the expectation of the random return (Morimura et al., 2010a;b; Bellemare et al., 2017; 2023). These methods have seen recent empirical successes in domains such as stratospheric balloon navigation (Bellemare et al., 2020), simulated race car control (Wurman et al., 2022), and algorithm discovery (Fawzi et al., 2022), as well as forming a core component of many successful agents in common simulated reinforcement learning benchmarks (Bellemare et al., 2013; Machado et al., 2018; Bellemare et al., 2017; Dabney et al., 2018b;a; Yang et al., 2019; Vieillard et al., 2020; Nguyen et al., 2021), often improving over agents that estimate only the expected return. Notably, the success of these distributional approaches has typically been observed in combination with deep neural networks, and it is commonly hypothesised that the benefits of the distributional approach stem from its interaction with non-linear function approximators such as deep neural networks (Bellemare et al., 2017; Imani and White, 2018; Dabney et al., 2021; Sun et al., 2022a), rather than for statistical reasons.

In this paper, however, we reach a surprising conclusion: Even in the tabular setting, there are many scenarios where *quantile temporal-difference learning* (QTD; Dabney et al., 2018b), a distributional RL algorithm which aims to learn quantiles of the return distribution, can more accurately estimate the expected return than classical temporal-difference learning (TD; Sutton, 1984; 1988) which predicts only the expected return.

To complement this core finding, we conduct novel theoretical analysis to establish what kinds of value predictions QTD converges to, and crucially how this depends on the number of quantiles that the algorithm estimates. We also examine how both TD and QTD trade-off between the variance of their updates, and their expected progress towards their asymptotic predictions. We find that when estimating a sufficient number of quantiles, QTD is able to converge to value predictions close to the true value function V^π, yet with individual updates that are guaranteed to be of bounded magnitude.

These insights lead to several testable hypotheses, which we use to conduct a further empirical study to better characterise domains in which QTD offers superior performance to TD, and vice versa, and find several common trends. In particular, we find that in environments with significant stochasticity, QTD often performs better (and in contrast, in (near-)deterministic environments, TD is clearly preferable), and that estimating a low number of quantiles may have adverse effects on the accuracy of QTD's predictions. By investigating a variant of QTD, we also find evidence that estimation specifically of the return distribution may lead to useful variance-reduction properties.

These findings have consequences for both theoreticians and practitioners. For the former, QTD represents a distinct fundamental approach to the problem of value prediction in RL, often with complementary performance to classical TD, and raises a range of open questions. For the latter, QTD for mean estimation can be considered as a plug-in alternative to TD, in tabular settings and beyond.

[1]DeepMind [2]Google Research, Brain team. Correspondence to: Mark Rowland <markrowland@deepmind.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

2. Background

We consider a Markov decision process, specified by a finite state space $\mathcal{X}$, action space $\mathcal{A}$, joint transition probabilities $P : \mathcal{X}\times\mathcal{A} \to \mathscr{P}(\mathcal{X}\times\mathbb{R})$ that specify for each $(x, a) \in \mathcal{X}\times\mathcal{A}$ a distribution over an immediate reward and next state, and a discount factor $\gamma \in [0, 1)$. When a policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ for selecting actions at each state is specified, each choice of an initial state $x_0 \in \mathcal{X}$ gives rise to a probability distribution over the trajectory $(X_t, R_t)_{t\geq 0}$, which we refer to as a Markov reward process (MRP). We introduce the notation $\mathbb{P}^{\pi}_{x_0}$ for the distribution of the trajectory, and $\mathbb{E}^{\pi}_{x_0}$ for the corresponding expectation operator.

The discounted return (or simply, the return) obtained along the trajectory $(X_t, R_t)_{t\geq 0}$ is defined as

$$\sum_{t\geq 0} \gamma^t R_t \,, \tag{1}$$

and encodes the utility of the trajectory to an agent interacting with the environment; higher returns are better. The value function $V^{\pi} : \mathcal{X} \to \mathbb{R}$ for a policy π is defined by

$$V^{\pi}(x) = \mathbb{E}^{\pi}_{x}\left[\sum_{t\geq 0} \gamma^t R_t\right] ,$$

for each $x \in \mathcal{X}$. That is, $V^{\pi}(x)$ is the mean return encountered along trajectories beginning at the state x. Estimating V^{π} from observed interactions with the environments is a fundamental problem in reinforcement learning, as it allows agents to both predict the effects of their actions, and to improve their policies.

2.1. Temporal-difference learning

Temporal-difference learning (TD; Sutton, 1984; 1988) is a family of algorithms that aim to learn an estimate of the value function V^{π}. We focus here on the simplest variant, TD(0). This algorithm maintains an estimate V of the value function, and incrementally updates these predictions in response to experience in the environment. Specifically, on observing a transition (x, r, x') generated by π, TD learning selects a learning rate $\alpha \in (0, 1]$, and performs the assignment

$$V(x) \leftarrow V(x) + \alpha(r + \gamma V(x') - V(x)) \,, \tag{2}$$

to update V. Under appropriate conditions, the estimate V converges to the true value function V^{π} with probability 1. The TD(0) update rule is a central method for tabular policy evaluation, and moreover, this update and its variants forms a core component of many deep reinforcement learning agents, including value-based approaches such as DQN and its descendants (Mnih et al., 2015), as well as actor-critic approaches such as A3C and its descendants (Mnih et al., 2016). For conciseness, throughout the paper we use "TD" to refer to this classical temporal-difference learning algorithm.

2.2. Quantile temporal-difference learning

In contrast to TD learning, which maintains an estimate of the expected return at each state, distributional RL algorithms (Bellemare et al., 2023) aim to predict the full probability distribution of the random return in Equation (1). Quantile temporal-difference learning (QTD Dabney et al., 2018b), in particular, maintains a *collection* of predictions at each state, denoted $((\theta(x, i))_{i=1}^{m} : x \in \mathcal{X})$, and has formed a core component of many deep reinforcement learning agents (Dabney et al., 2018a; Yang et al., 2019; Bodnar et al., 2020; Bellemare et al., 2020; Wurman et al., 2022; Fawzi et al., 2022). The intention, in contrast to having $V(x)$ directly approximate the mean return from x in TD learning, is to have $\theta(x, i)$ approximate the τ_i-quantile of the distribution of the random return in Equation (1), with $\tau_i = \frac{2i-1}{2m}$, for $i = 1, \ldots, m$; see Figure 1 for an illustration. We write QTD(m) to denote the instantiation of QTD with m quantiles.

In analogy with TD learning, upon observing a transition (x, r, x') generated by π, QTD updates all estimates $(\theta(x, i))_{i=1}^{m}$ at state x by selecting a learning rate $\alpha \geq 0$, and performing the assignments

$$\theta(x, i) \leftarrow \theta(x, i) + \alpha\Big(\tau_i - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[r + \gamma\theta(x', j) - \theta(x, i) < 0]\Big) \,, \tag{3}$$

for all $i = 1, \ldots, m$. This algorithm differs from TD in a few important ways. First, each prediction $\theta(x, i)$ is updated differently, due to the presence of the parameter τ_i in the update. Second, the update depends only on the *sign* (not magnitude) of the temporal-difference errors appearing in Equation (3), meaning that the update magnitude is bounded, in contrast to those of TD. The form of the update itself is motivated through the quantile regression loss (Koenker and Bassett, 1978; Koenker, 2005); see Rowland et al. (2023) and Bellemare et al. (2023) for further background and theory regarding QTD, and Appendix A for an overview and discussion of computational considerations.

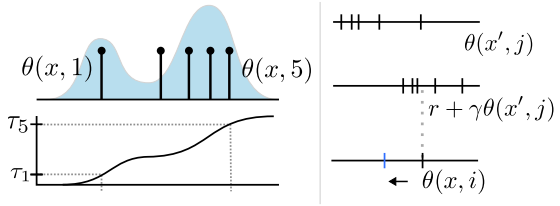

Figure 1. Left: Example return distribution at a state x in light blue, with QTD quantile predictions $(\theta(x, i))_{i=1}^{5}$ ($m = 5$). Exact predictions correspond to the quantiles indicated on the CDF below. Right: Illustration of computation of the update to $\theta(x, i)$ (from black to blue marker) upon observing a transition (x, r, x').

3. Quantile temporal-difference learning for mean-return estimation

Quantile temporal-difference learning estimates the return distribution at a state x with the discrete distribution supported at the learnt quantile values $(\theta(x,i))_{i=1}^m$:

$$\sum_{i=1}^{m} \frac{1}{m} \delta_{\theta(x,i)} ;$$

see Figure 1. A natural estimator for value at state x is therefore obtained by extracting the mean of this approximate distribution, by averaging these quantiles:

$$\frac{1}{m} \sum_{i=1}^{m} \theta(x,i) . \tag{4}$$

This is the estimator of value typically used in applications combining QTD with deep reinforcement learning. The approach of averaging certain quantile estimators to approximate the mean of a distribution in fact dates back to at least the work of Daniell (1920) and Mosteller (1946), with Gastwirth (1966) observing that this approach to estimation provides competitive relative efficiency across a wide variety of distributions, including those with heavy tails, where the usual sample-average mean estimator can be inefficient.

Thus, although not originally designed with this connection in mind, QTD naturally combines this approach to mean estimation with the notion of bootstrapping in reinforcement learning. Given the motivation above, we might conjecture that QTD provides an approach to value estimation that is effective across a wide range of environments, particularly those with heavy-tailed reward distributions. For concreteness, the QTD algorithm for value estimation is presented in Algorithm 1. The additional variables θ' are used to avoid issues when $x'_t = x_t$; in such cases, this means that the for-loop over the quantile index i can be performed in any order (or in parallel) without affecting the result of the algorithm.

Algorithm 1 QTD(m) for value estimation.

Require: Initial quantile estimates $((\theta(x,i))_{i=1}^m : x \in \mathcal{X})$, learning rate α, number of updates T.

for $t = 1, \dots, T$ **do**
  Observe transition (x_t, r_t, x'_t).
  for $i = 1, \dots, m$ **do**
    Set $\theta'(x_t,i) \leftarrow \theta(x_t,i) + \alpha\Big(\tau_i - \frac{1}{m}\sum_{j=1}^m \mathbb{1}[\theta(x_t,i) - r_t - \gamma\theta(x'_t,j) < 0]\Big)$
  end for
  Set $\theta(x_t,i) \leftarrow \theta'(x_t,i)$ for $i = 1, \dots, m$
end for
return $(\frac{1}{m}\sum_{i=1}^m \theta(x,i) : x \in \mathcal{X})$

As an initial comparison between TD and QTD for value estimation, we compare mean-squared error for value estimation on a suite of nine simple MRPs. Full details for replication are provided in Appendix C, with crucial details for the comparisons given here. The structure of the MRPs is given by the Cartesian product of three levels of stochasticity in transition structure:

- Deterministic cycle structure;
- Sparse stochastic transition structure (sampled from a Garnet distribution;
- Dense stochastic transition structure (sampled from Dirichlet$(1, \dots, 1)$ distributions); Archibald et al., 1995);

together with three levels of stochasticity in reward structure:

- Deterministic rewards;
- Gaussian (variance 1) rewards;
- Exponentially distributed (rate 1) rewards.

We focus on the use of constant learning rates throughout training, as is commonly the case in practice, and sweep across a variety of learning rates for both methods. We run both TD and QTD (using 128 quantiles) with a variety of learning rates, and measure the mean-squared error to the true value function after 1,000 updates via online interaction with the environments. The results of the sweep over learning rates are displayed in Figure 2; in this experiment and all that follow, each run was repeated 1,000 times, and the (narrow) confidence bands displayed are obtained via a measurement of ± 2 times the empirical standard error.

As expected, in the environments with the heaviest-tailed rewards, QTD obtains a lower mean-squared error than TD. Interestingly, this is also the case in environments with stochasticity only in the transition dynamics, and deterministic rewards. To more easily visualise the extent of these improvements, and to check the robustness of this improvement to the number of updates undertaken by the algorithms, we plot the optimal MSE obtained by QTD as a proportion of that obtained by TD in Figure 3, as a function of the number updates completed by each algorithm. This preliminary experiment has already yielded a perhaps surprising conclusion:

> *Even in the tabular setting, QTD, a distributional reinforcement learning algorithm, can outperform classical TD learning in estimating expected returns.*

In addition to obtaining superior performance relative to TD when optimising over learning rates, Figure 2 also indicates that performance degradation due to a larger-than-optimal learning rate is considerably less severe with QTD than with TD in these environments.

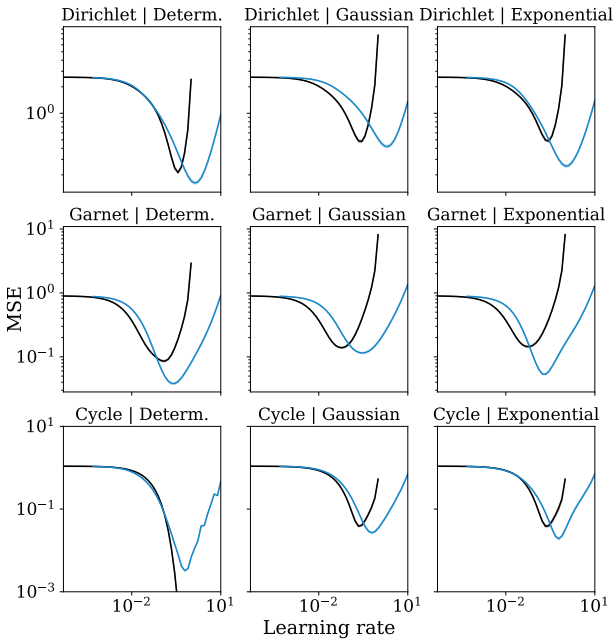

Figure 2. Mean-squared error against learning rate for TD (black) and QTD(128) (blue), on environments with Dirichlet transition structure (top), Garnet transition structure (middle), and deterministic cycle structure (bottom), and deterministic rewards (left), Gaussian rewards (centre), and exponential rewards (right).

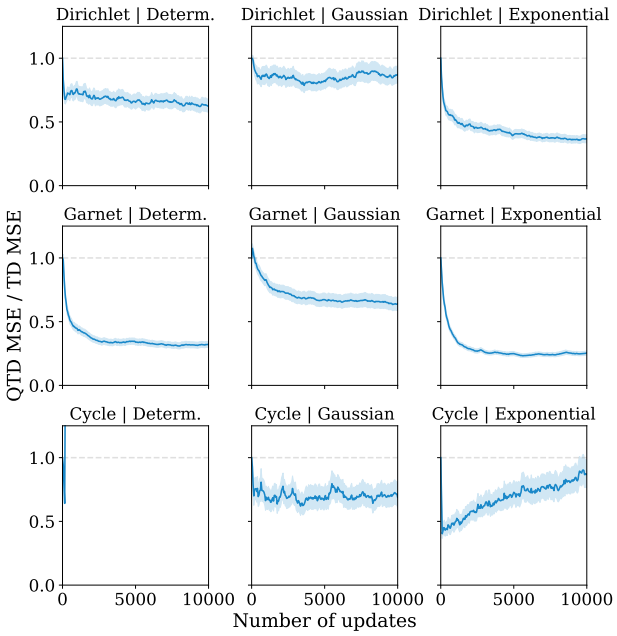

Figure 3. Improvement of QTD(128) over TD in mean-squared error against number of updates for all environments in Figure 2.

Importantly, however, we also note that for the deterministic environment in the suite, the performance of TD is far superior to QTD. In this case, the TD algorithm is able to very accurately approximate the value function, since the update in Equation 2 is essentially implementing exact asynchronous dynamic programming when $\alpha = 1$.

The results above have shown that in some sense, QTD has a complementary performance profile to TD, viewed as algorithms for value estimation, and that the stochaticity of the environment is one important factor in determining the relative performance of QTD and TD. What else can be said about the performance of QTD in comparison with TD? We address this questions in two ways. First, we develop the theory of QTD for mean estimation in Section 4, establishing asymptotic guarantees on the quality of the value predictions that the algorithm makes. Second, we conduct further empirical investigations in Section 5, aiming to develop a more nuanced understanding of the relative performance of QTD and TD in practice.

4. Theoretical analysis

For the TD learning update rule in Section 2.1, it is known that under mild conditions on the reward distributions of the MRP, learning rates, and frequency that each state x is updated, the predictions V converge to V^π with probability 1 (Watkins, 1989; Watkins and Dayan, 1992; Dayan, 1992; Dayan and Sejnowski, 1994; Tsitsiklis, 1994; Jaakkola et al., 1994; Bertsekas and Tsitsiklis, 1996). This core convergence result justifies the use of TD for policy evaluation.

Rowland et al. (2023) show that under even milder conditions, QTD(m) also converges with probability 1 *. However, in general, the estimate of expected returns V_m^{QTD} extracted from a point of convergence θ_m^{QTD} of QTD via Equation (4) is *not* exactly equal to V^π. Intuitively, this stems from the fact that the algorithm is not estimating the mean of the return distribution directly, but rather a finite collection of quantiles, and this information is insufficient to exactly reconstruct the mean of the distribution in question.

Based on this observation, we bound the expected error of the (random) value estimator $\hat{V}$ obtained from Algorithm 1 as follows:

$$\mathbb{E}[\|\hat{V} - V^\pi\|] \leq \underbrace{\mathbb{E}[\|\hat{V} - V_m^{\text{QTD}}\|]}_{\text{Finite-sample error}} + \underbrace{\|V_m^{\text{QTD}} - V^\pi\|}_{\text{Fixed-point error}} . \quad (5)$$

The precise norm is unimportant here; the main aim is to highlight the role played by fixed-point error and finite-sample error in the overall error incurred by the QTD value estimator. We now compare QTD and TD with each of these terms in mind, beginning with fixed-point error.

*QTD may converge to a *set* of fixed points, rather than a unique fixed point as with TD; the discussion above refers to any point in this set of fixed points.

4.1. Fixed-point error

As noted above, TD incurs zero fixed-point error, as its point of convergence is precisely V^π. However, this is generally not true of QTD. Nevertheless, it is possible to bound the fixed-point error of QTD as a function of the number of quantiles estimated by QTD in many cases. The following result is a straightforward consequence of the fixed-point analysis of Rowland et al. (2023). Proofs of results stated in the main paper are provided in Appendix B.

Proposition 4.1. For an MRP with all reward distributions supported on $[R_{\min}, R_{\max}]$, any convergence point θ_m^{QTD} of QTD(m) with corresponding value function estimate $V_m^{\text{QTD}} = (\frac{1}{m}\sum_{i=1}^m \theta_m^{\text{QTD}}(x,i) : x \in \mathcal{X})$ satisfies

$$\|V_m^{\text{QTD}} - V^\pi\|_\infty \leq \frac{R_{\max} - R_{\min}}{2m(1-\gamma)^2}\,.$$

This guarantees that in the case of bounded reward distributions, we can ensure that the fixed points of QTD provide arbitrarily accurate value function estimates, as long as m is taken to be sufficiently large relative to the scale of the support of the reward distributions.

Remark 4.2. The form of this approximation error is easily interpreted; for a general distribution supported on $[R_{\min}/(1-\gamma), R_{\max}/(1-\gamma)]$ (as the return distribution at x is under the conditions of Proposition 4.1), with mean μ and quantile function F^{-1}, we have

$$\mu = \int_0^1 F^{-1}(\tau)\,\mathrm{d}\tau \approx \sum_{i=1}^m \frac{1}{m} F^{-1}\Big(\frac{2i-1}{2m}\Big)\,. \tag{6}$$

That is, estimating the mean with a finite number of quantiles can be understood as a midpoint-quadrature-based approximation to the true mean. From this point of view, a linear dependence of the error on the range $(R_{\max} - R_{\min})/(1-\gamma)$ of the integrand in Equation (6), and a dependence $1/m$ on the number of quadrature points m are to be expected. The additional factor of $(1-\gamma)^{-1}$ in the bound stems from the fact that the estimate is obtained from the fixed point of a bootstrapping procedure, in which errors accumulate at each stage. Since only a finite number m of quantiles are estimated at each state, the remaining information about the return distributions is thrown away, and this results in an accumulation of error each time the update in Equation (3) is applied. This bears a relationship to the notion of Bellman closedness in distributional RL (Rowland et al., 2019; Bellemare et al., 2023), and is analogous to the compounding of error under linear function approximation (Tsitsiklis and Van Roy, 1997).

We now develop this analysis further, obtaining results for environments with unbounded rewards. We state a bound for the important case of sub-Gaussian rewards below, which follows as a consequence of a much more general bound given by Proposition B.1.

Proposition 4.3. Consider an MRP with all reward distributions having means in $[R_{\min}, R_{\max}]$, and all sub-Gaussian with parameter σ^2, so that $\mathbb{E}_x^\pi[\exp(\lambda(R - \mathbb{E}_x^\pi[R]))] \leq \exp(\lambda^2\sigma^2/2)$, for all $\lambda \in \mathbb{R}$ and $x \in \mathcal{X}$. Then for the value function estimate V_m^{QTD} obtained from any convergence point θ_m^{QTD} of QTD(m) via Equation (4), we have

$$\|V_m^{\text{QTD}} - V^\pi\|_\infty \leq \frac{1}{(1-\gamma)m} \times \left(\frac{R_{\max} - R_{\min} + 2\sigma\sqrt{2\log(2m)}}{2(1-\gamma)} + \frac{\sigma}{\sqrt{2\log(2m)}}\right).$$

We also state a non-quantitative result applicable to any MDP for which the problem of mean return estimation is well defined.

Proposition 4.4. Consider an MDP with all reward distributions having finite mean. Then for the value function estimate V_m^{QTD} obtained from any convergence point θ_m^{QTD} of QTD(m) via Equation (4), we have $\|V_m^{\text{QTD}} - V^\pi\|_\infty \to 0$ as $m \to \infty$.

This analysis shows that even with unbounded reward distributions, the approximation error of the fixed points of QTD can still be made arbitrarily small by increasing m, with a slightly slower rate (relative to the bounded-reward case) of $O(m^{-1}\sqrt{\log(m)}(1-\gamma)^{-2})$ in the case of sub-Gaussian rewards; in general, the heavier the tails of the reward distributions, the slower the convergence may be.

4.2. Expected updates and variance

The analysis of the previous section shows that QTD (with a large enough number of quantiles) incurs low fixed-point error, but does not suggest how its finite-sample performance may compare to that of TD, and specifically in which kinds of environments it may outperform TD. To make progress on this question, we return to the other term in Inequality (5), and in particular consider how the updates of TD and QTD contribute to this quantity. We begin by considering the updates of TD.

TD update decomposition. The right-hand side of the TD learning update in Equation (2) can be rewritten as

$$\begin{aligned} &V(x) + \alpha(r + \gamma V(x') - V(x)) \\ =&(1-\alpha)V(x) + \\ &\alpha(\underbrace{(T^\pi V)(x)}_{\text{Expected update}} + \underbrace{(r + \gamma V(x') - (T^\pi V)(x))}_{\text{Mean-zero noise}})\,, \end{aligned}$$

where $(T^\pi V)(x) = \mathbb{E}_x^\pi[R_0 + \gamma V(X_1)]$ is the classical dynamic programming operator. This decomposition is central

	Fixed-point bias	Update variance	Expected update magnitude
TD	0	Unbounded*	$\propto$ Bellman error
QTD	$\widetilde{\mathcal{O}}(1/m)^{**}$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Table 1. Trade-offs made by TD and QTD along various axes. *In general, TD update variance may be unbounded, though there are certain situations in which it is not; see text for further discussion. **For sub-Gaussian reward distributions. $\tilde{O}$ denotes the possible dropping of polylog factors in m.

to the analyses of TD cited above, and highlights that the learning rate α balances two requirements: a large learning rate increases the expected update towards $(T^\pi V)(x)$, increasing the contraction towards the fixed point V^π of T^π, but also amplifies the mean-zero noise. Note also that the magnitude of the noise is potentially unbounded (if there are unbounded rewards, or if the value estimate V grows large), and that the distance of the expected update $(T^\pi V)(x)$ also grows in magnitude with V.

The key to obtaining good performance, and low finite-update error, from TD is therefore selecting a learning rate that balances the tension between these two considerations. These links between temporal-difference learning and dynamic programming are well understood (see e.g. Jaakkola et al. (1994); Tsitsiklis (1994); Bertsekas and Tsitsiklis (1996)), and this specific trade-off has been previously quantified under a variety of formalisms; see the work of Kearns and Singh (2000) for the *phased* setting, and Even-Dar and Mansour (2003) for the synchronous and online settings.

QTD update decomposition. In analogy, we can also decompose the QTD update in Equation (3) into an expected update and mean-zero noise; this approach is central to the convergence analysis of Rowland et al. (2023). In particular, the right-hand side of Equation (3) can be decomposed as follows:

$$\alpha\Big(\tau_i - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[r + \gamma\theta(x', j) < \theta(x, i)]\Big)$$
$$= \alpha\Big(\underbrace{\tau_i - \mathbb{P}_x^\pi(\Delta_{iJ}(x, R_0, X_1) < 0)}_{\text{Expected update}} +$$
$$\underbrace{\mathbb{P}_x^\pi(\Delta_{iJ}(x, R, X') < 0) - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[\Delta_{ij}(x, r, x') < 0]}_{\text{Mean-zero noise}}\Big).$$

where we write $\Delta_{ij}(x, r, x') = r + \gamma\theta(x', j) - \theta(x, i)$. Rowland et al. (2023) show that following these expected updates leads to the points of convergence for QTD, and we therefore have a similar tension as described in TD, between an expected update that moves us towards the points of convergence, and noise that may perturb this progress.

A central distinction between this decomposition for TD, and for QTD, is that in QTD both expected update and noise are bounded by 1, in stark contrast to the potentially unbounded terms in the TD update, which may grow in proportion with the value function norm $\|V\|_\infty$. This suggests that QTD may tolerate higher step sizes than TD in stochastic environments, and also that as the level of stochasticity increases, due to higher-variance/heavier-tailed rewards, the performance of QTD may be more resilient than that of TD. Conversely, in near-deterministic environments, since the expected update magnitude of QTD is effectively independent of the magnitude of the update error, we may expect poorer performance than TD, which is able to make updates in proportion to the level of error. A summary of the comparison points highlighted between TD and QTD in this section is given in Table 1.

5. Further empirical analysis

The theoretical analysis in the previous section has elucidated several salient differences between TD and QTD as policy evaluation algorithms; we now seek to compare these methods empirically, and test the predictions made in light of the analysis in the earlier sections.

5.1. Heavy-tailed rewards

As alluded to above, the sensitivity of TD updates to the magnitude of prediction errors makes it difficult to average out heavy-tailed noise, and we hypothesise that in cases of extremely heavy-tailed noise, QTD should strongly outperform TD. To this end, we extend the example environments from Figure 2, with t_2-distributed rewards; these are exceptionally heavy-tailed rewards, with infinite variance. The results of QTD and TD in these environments are displayed in Figure 4, with QTD providing substantial improvements in MSE. A plot of MSE against learning rates is provided in Appendix D.2.

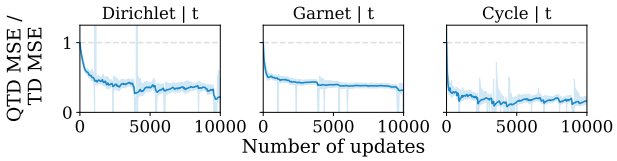

Figure 4. Relative improvement of QTD(128) over TD in mean-squared error against number of updates for all transition structures in Figure 2, with t_2-distributed rewards.

5.2. Low numbers of quantiles

Propositions 4.1, 4.3, and 4.4 suggest that for low numbers of quantiles m, the fixed-point bias of QTD may dominate the error decomposition described above, meaning that it

may be outperformed by TD in certain environments. Figure 5 illustrates such a case, under the same experimental set-up as earlier in the section; MSE is poor with low values of m for all learning rates, and comparable performance to TD is recovered by increasing m.

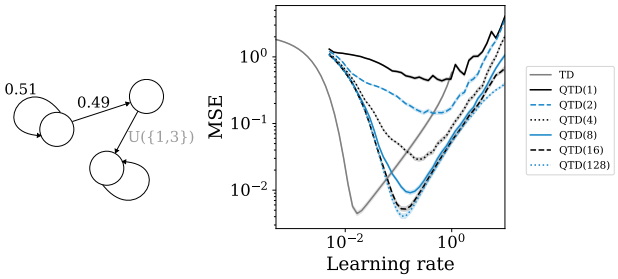

Figure 5. An example environment (left) where low values of m induce particularly high fixed-point bias. learning rate vs. mean-squared error for QTD(m) with varying m (right).

What makes QTD(m) with m low behave so poorly in this example? It is precisely due to the fact that the average of a low number of quantiles in this domain is quite different from the mean, as mentioned in Remark 4.2. This serves to illustrate cases where large numbers of quantiles are necessary for accurate predictions, as the theory in Section 4 suggests. We also include results on the main suite of environments for QTD(1) and QTD(16) in Appendix D.3, for comparison with the results obtained for QTD(128) above. QTD(1) is outperformed by TD in several environments, as the experiment in Figure 5 suggests may be the case. On the other hand, the performance of QTD(16) is broadly in line with that of QTD(128); speaking pragmatically, we have found that using on the order of tens of quantiles is generally sufficient in practice.

5.3. Varying reward scales

Given our previous observations that TD outperforms QTD in deterministic environments, and that QTD tends to outperform TD in environments with significant stochasticity, we run an additional comparison to investigate the levels of stochasticity required to see benefits from QTD. In Figure 2, we see advantages to QTD in all environments with stochastic transition structure, but a clear difference in performance in passing from the environment with cycle transition structure and Gaussian reward (centre-bottom) to the same transition structure with deterministic rewards (bottom-left). In Figure 6, we plot the relative performance of QTD(128) and TD in the environment with cycle transition structure, and Gaussian rewards of varying levels of standard deviation. The results show that at low levels of reward noise, the performance of TD is far superior to QTD, as in the purely deterministic case, with the relative performance of QTD improving monotonically as a function of the standard deviation of the reward noise.

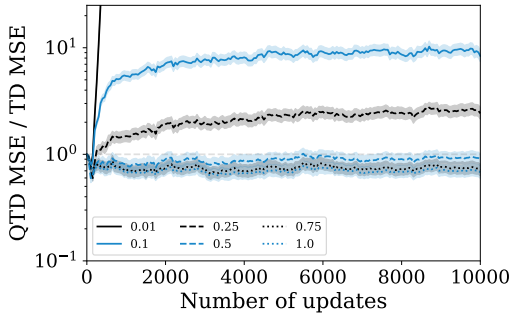

Figure 6. Performance of QTD(128) relative to TD, with optimal learning rates, on the environment with deterministic cycle transition structure, and Gaussian rewards of varying standard deviation, indicated in the legend.

5.4. An ablation: Pseudo-quantile temporal-difference learning

We have motivated QTD theoretically as an effective algorithm for tabular policy evaluation, and we have also seen this borne out empirically. We have described its contrasting performance profile to TD, and noted its properties of (i) bounded-magnitude updates, and (ii) controllable fixed-point error. Taking a step back, a natural question to ask is: are there further nuances to the particular form of the QTD update which make it an effective algorithm? To investigate this question further, in this section we study a new algorithm for tabular policy evaluation, *pseudo-quantile temporal-difference learning* (PQTD), which uses the same form of quantile updates as QTD, though does *not* aim to learn quantiles of the return distribution. Our goal is to understand the role played by these two components of QTD in forming an effective policy evaluation algorithm.

In particular, motivated by Achab's (2020) study of the one-step random return $R + \gamma V^\pi(X')$ (see also Achab and Neu (2021), Achab et al. (2022), and Achab et al. (2023)), PQTD aims to learn the quantiles of the distribution of these random variables, rather than those of the usual return distribution. The approach is presented in Algorithm 2. The distinction from QTD is that the targets in the quantile regression update are constructed from the mean-return estimate at the next state, rather than from the quantile estimates themselves; the learnt quantile estimates therefore reflect only the randomness resulting from a single step of environment interaction. This is also motivated by the approach of two-hot encoded categorical value learning in recent deep RL applications (Schrittwieser et al., 2020; Hessel et al., 2021; Hafner et al., 2023), which can be interpreted as a one-step version of categorical distributional RL (Bellemare et al., 2017).

The results of running PQTD on the same suite of environ-

Algorithm 2 PQTD update.

Require: Quantile estimates $((\theta(x,i))_{i=1}^m : x \in \mathcal{X})$, observed transition (x, r, x'), learning rate α.

for $i = 1, \dots, m$ **do**

  $\theta'(x,i) \leftarrow \theta(x,i) + \alpha\Big(\tau_i - \mathbb{1}[\theta(x,i) - r - \gamma \frac{1}{m} \sum_{j=1}^m \theta(x',j) < 0]\Big)$

end for

Set $\theta(x,i) \leftarrow \theta'(x,i)$ for $i = 1, \dots, m$

return $((\theta(y,i))_{i=1}^m : y \in \mathcal{X})$

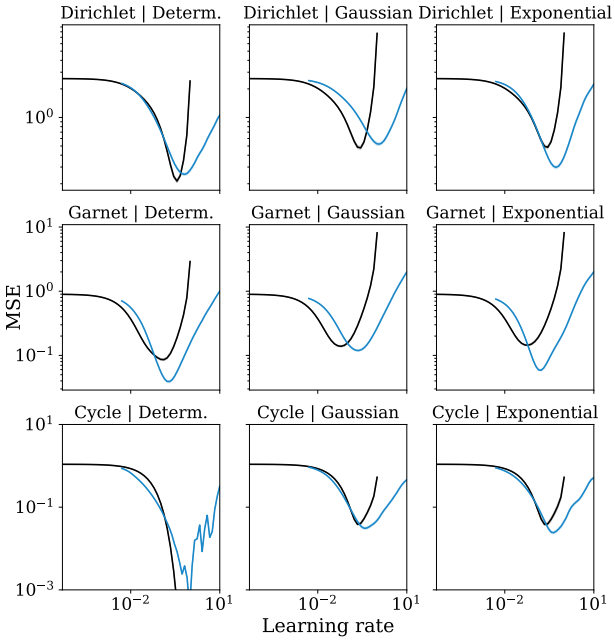

Figure 7. Mean-squared error against learning rate for TD (black) and PQTD(128) (blue).

ments as reported in Figure 2 are given in Figure 7, with improvements at optimised learning rates across a range of number of updates displayed in Figure 8. Overall, similar behaviour is observed with PQTD as with QTD: larger learning rates to TD are preferred, and the approach tends to work best in the presence of high environment stochasticity. However, the level of performance obtained is generally somewhat worse than QTD, and worse than TD in several stochastic environments too; this discrepancy provides a useful opportunity to understand the success of QTD better.

In particular, considering the cycle environment with Gaussian reward noise, the fixed-point bias for both QTD and PQTD is in fact zero in this case; for readers familiar with distributional dynamic programming (Bellemare et al., 2023), intuitively this follows from symmetry of the one-step target distributions, meaning that the average of the learnt quantiles is equal to the mean. Our earlier decomposition of the error therefore suggests that the discrepancy

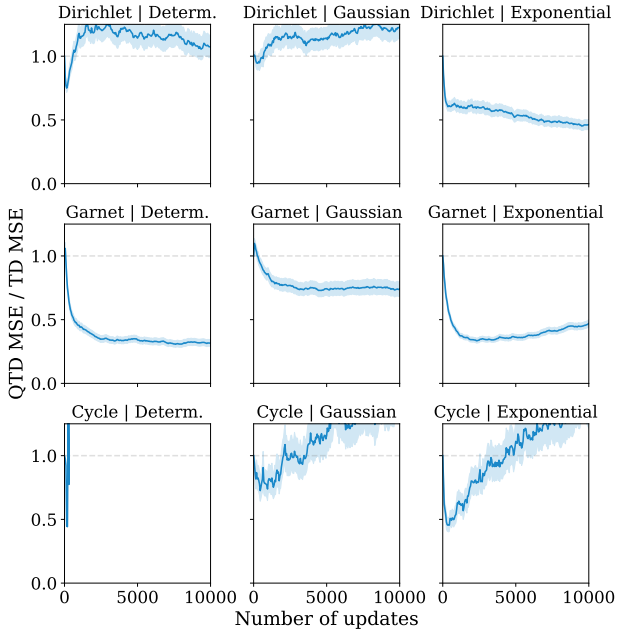

Figure 8. Relative improvement in MSE of PQTD over TD, with numbers of updates ranging from 0 to 10,000.

between QTD and PQTD must arise from finite-sample error, and points to differences in the update variance between the algorithms. Our empirical observations concur with this conjecture, with PQTD updates often having variance several times greater than those of QTD around the points of convergence. The form of the update for both QTD and PQTD is also informative; the term multiplying the learning rate in Algorithm 2 can take on only the values τ_i or $1 - \tau_i$, whereas the averaging that occurs in the corresponding QTD update allows for significantly lower-magnitude updates, and hence potentially lower variance.

This finding suggests that a strength of QTD for value estimation is not only its bounded-magnitude updates, but the fact that variance of these updates is often significantly better than the bounds alone suggest, and that specifically learning the distribution of the full return can have beneficial variance-reduction properties in temporal-difference learning.

6. Related work

Mean estimation with quantiles. The approach of estimating a location parameter by averaging quantiles dates back at least to Daniell (1920), who investigated the non-uniform averaging of order statistics to estimate a one-dimensional location parameter. Mosteller (1946) developed this line of work further, investigating the statistical properties of averages of quantile estimates in greater detail. Interestingly, several proposals for which quantile levels should be aver-

aged were made in this work, including the levels $\tau_i = \frac{2i-1}{2m}$ used by QTD, though without theoretical justification. Gastwirth (1966) also studied the efficiency of a mean estimator based on averaging of three specific quantiles for symmetric distributions with varying levels of heavy-tailedness. Huber (1964) proposed using smoothed versions of quantile losses for location estimation. See also Andrews et al. (1972) for a broader review of robust approaches to location estimation. Online estimation of quantiles via incremental algorithms also has a long history; quantile estimation (in the supervised learning setting) is one of the examples provided by Robbins and Monro (1951) in their work introducing the field of stochsatic approximation.

Deep quantile temporal-difference learning. In addition to the original QTD algorithm (Dabney et al., 2018b), recent theoretical developments (Lhéritier and Bondoux, 2022; Rowland et al., 2023), and extensions in the context of deep reinforcement learning (Dabney et al., 2018a; Yang et al., 2019), several architectural innovations specifically exploiting neural network function approximation have been proposed to avoid the quantile-crossing problem when combining QTD with neural function approximation (Zhou et al., 2020; Luo et al., 2021; Théate et al., 2021).

Distributional reinforcement learning algorithms. In this paper, we have focused on quantile temporal-difference learning, a particular instance of a distributional reinforcement learning algorithm. Other distributional reinforcement learning algorithms include categorical temporal-difference learning (CTD; Bellemare et al., 2017; Rowland et al., 2018), maximum-mean discrepancy-based methods (Nguyen et al., 2021), methods using distributional representations based on mixtures of Gaussians (Barth-Maron et al., 2018), and methods using Sinkhorn divergences (Sun et al., 2022b). It is interesting to contrast the finding that QTD is a strong algorithm for tabular policy evaluation, with properties that *complement* those of TD, with prior findings relating to CTD (Rowland et al., 2018; Lyle et al., 2019; Bellemare et al., 2023). In contrast to QTD, this prior work showed that in many circumstances, CTD behaves *identically* to TD for mean estimation, and so offers no additional benefit, or complementary profile of performance.

Robust approaches to TD learning and optimisation. A variety of approaches to robust and regularised variants of TD learning have been considered previously (Bossaerts et al., 2020; Lu and Giannakis, 2021; Meyer, 2021; Klima et al., 2019; Ghiassian et al., 2020; Liu et al., 2012; Manek and Kolter, 2022). Bounded updates naturally arise from the QTD learning algorithm; bounded gradients are also commonly encountered in deep learning as a heuristic approach to stabilising optimisation through clipping (Mikolov, 2012; Pascanu et al., 2013), as well as in fundamental optimisation algorithms (Riedmiller and Braun, 1993).

7. Conclusion

In this paper, we have shown that QTD can be viewed as a fundamental algorithm for policy evaluation, with complementary properties to the classical approach to temporal-difference learning. The theoretical and empirical analysis, as well as the introduction and study of the related algorithm PQTD, has given indications as to which kinds of environments we might expect one approach to improve over the other. We emphasise that these findings are of course not exhaustive, and we expect there to be significant value in further empirical investigation of QTD as a tabular policy evaluation algorithm, as well as analysis of variants incorporating aspects such as multi-step returns, off-policy corrections, and function approximation, all of which interact in various ways with the complementary trade-offs made by TD and QTD between fixed-point error, variance and expected update magnitude (White and White, 2016; Mahmood et al., 2017; Rowland et al., 2020). Precise finite-sample bounds on performance are also a natural direction for future work.

These findings are also pertinent to the overarching questions as to where exactly the benefits of distributional RL stem from. Common hypotheses have often focused on the interaction between distributional predictions and non-linear function approximation, with mechanisms such as improved representation learning, prevention of rank collapse, and improved loss landscapes being proposed. This work highlights that even in risk-neutral tabular settings, there are benefits to taking a distributional approach to reinforcement learning, and opens up several directions of research to understand the role of distributional RL as a core technique in reinforcement learning. Historically, distributional RL algorithms have often been evaluated in (near-)deterministic environments; this paper also supports the idea that by evaluating algorithms on a wider range of environments, we may obtain a more nuanced view of the strengths and weaknesses of the algorithms at play. Above all, this paper aims to show that distributional reinforcement learning has a fundamental role in developing algorithms that complement our existing approaches to core tasks such as policy evaluation; to estimate the mean, it can pay to estimate the full distribution.

Acknowledgements

We thank David Abel for detailed comments on an earlier draft, and the reviewers & area chair for their helpful comments on the paper. The experiments in this paper were undertaken using the Python 3 language, and made use of the NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), and Matplotlib (Hunter, 2007) libraries.

References

Mastane Achab. *Ranking and risk-aware reinforcement learning*. PhD thesis, Institut Polytechnique de Paris, 2020.

Mastane Achab and Gergely Neu. Robustness and risk management via distributional dynamic programming. *arXiv*, 2021.

Mastane Achab, Reda Alami, Yasser Abdelaziz Dahou Djilali, Kirill Fedyanin, Eric Moulines, and Maxim Panov. Distributional deep Q-learning with CVaR regression. In *Deep Reinforcement Learning Workshop, NeurIPS*, 2022.

Mastane Achab, Reda Alami, Yasser Abdelaziz Dahou Djilali, Kirill Fedyanin, and Eric Moulines. One-step distributional reinforcement learning. *arXiv*, 2023.

David F. Andrews, Peter J. Bickel, Frank R. Hampel, Peter J. Huber, William H. Rogers, and John W. Tukey. *Robust Estimates of Location: Survey and Advances*. Princeton University Press, 1972.

T. W. Archibald, K. I. M. McKinnon, and L. C. Thomas. On the generation of Markov decision processes. *Journal of the Operational Research Society*, 46(3):354–361, 1995.

Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. In *Proceedings of the International Conference on Learning Representations*, 2018.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2017.

Marc G. Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C. Machado, Subhodeep Moitra, Sameera S. Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional reinforcement learning*. MIT Press, 2023.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

Cristian Bodnar, Adrian Li, Karol Hausman, Peter Pastor, and Mrinal Kalakrishnan. Quantile QT-Opt for risk-aware vision-based robotic grasping. In *Robotics: Science and Systems*, 2020.

Peter Bossaerts, Shijie Huang, and Nitin Yadav. Exploiting distributional temporal difference learning to deal with tail risk. *Risks*, 8(4):113, 2020.

David Cheikhi and Daniel Russo. On the statistical benefits of temporal difference learning. *arXiv*, 2023.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018a.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018b.

Will Dabney, André Barreto, Mark Rowland, Robert Dadashi, John Quan, Marc G. Bellemare, and David Silver. The value-improvement path: Towards better representations for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

P. J. Daniell. Observations weighted according to order. *American Journal of Mathematics*, 42(4):222–236, 1920.

Peter Dayan. The convergence of TD(λ) for general λ. *Machine learning*, 8(3-4):341–362, 1992.

Peter Dayan and Terrence J. Sejnowski. TD(λ) converges with probability 1. *Machine Learning*, 14(3):295–301, 1994.

Eyal Even-Dar and Yishay Mansour. Learning rates for Q-learning. *Journal of Machine Learning Research*, 5(1), 2003.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

Joseph L Gastwirth. On robust procedures. *Journal of the American Statistical Association*, 61(316):929–948, 1966.

Sina Ghiassian, Andrew Patterson, Shivam Garg, Dhawal Gupta, Adam White, and Martha White. Gradient temporal-difference learning with regularized corrections. In *Proceedings of the International Conference on Machine Learning*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv*, 2023.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Matteo Hessel, Ivo Danihelka, Fabio Viola, Arthur Guez, Simon Schmitt, Laurent Sifre, Theophane Weber, David Silver, and Hado van Hasselt. Muesli: Combining improvements in policy optimization. In *Proceeding of the International Conference on Machine Learning*, 2021.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International Conference on Machine Learning*, pages 2157–2166. PMLR, 2018.

Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

Michael J. Kearns and Satinder Singh. Bias-variance error bounds for temporal difference updates. In *Proceedings of the Conference on Learning Theory*, 2000.

Richard Klima, Daan Bloembergen, Michael Kaisers, and Karl Tuyls. Robust temporal difference learning for critical domains. *arXiv*, 2019.

Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978.

Harold Kushner and G. George Yin. *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media, 2003.

Alix Lhéritier and Nicolas Bondoux. A Cramér distance perspective on quantile regression based distributional reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2022.

Bo Liu, Sridhar Mahadevan, and Ji Liu. Regularized off-policy TD-learning. *Advances in Neural Information Processing Systems*, 2012.

Qin Lu and Georgios B. Giannakis. Robust and adaptive temporal-difference learning using an ensemble of Gaussian processes. *arXiv*, 2021.

Yudong Luo, Guiliang Liu, Haonan Duan, Oliver Schulte, and Pascal Poupart. Distributional reinforcement learning with monotonic splines. In *Proceedings of the International Conference on Learning Representations*, 2021.

Clare Lyle, Marc G. Bellemare, and Pablo Samuel Castro. A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the Arcade Learning Environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Ashique Rupam Mahmood, Huizhen Yu, and Richard S Sutton. Multi-step off-policy learning without importance sampling ratios. *arXiv*, 2017.

Gaurav Manek and J Zico Kolter. The pitfalls of regularization in off-policy td learning. In *Advances in Neural Information Processing Systems*, 2022.

Dominik Jakob Meyer. *Accelerated Gradient Algorithms for Robust Temporal Difference Learning*. PhD thesis, Technische Universität München, 2021.

Tomáš Mikolov. *Statistical language models based on neural networks*. PhD thesis, Brno University of Technology, Faculty of Information Technology, 2012.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings*

of the International Conference on Machine Learning, 2016.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return density estimation for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2010a.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010b.

Frederick Mosteller. On some useful "inefficient" statistics. *Annals of Mathematical Statistics*, 17:377–408, 1946.

Thanh Tang Nguyen, Sunil Gupta, and Svetha Venkatesh. Distributional reinforcement learning via moment matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, 2013.

Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591. IEEE, 1993.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Mark Rowland, Marc G. Bellemare, Will Dabney, Rémi Munos, and Yee Whye Teh. An analysis of categorical distributional reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2018.

Mark Rowland, Robert Dadashi, Saurabh Kumar, Rémi Munos, Marc G. Bellemare, and Will Dabney. Statistics and samples in distributional reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2019.

Mark Rowland, Will Dabney, and Rémi Munos. Adaptive trade-offs in off-policy learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2020.

Mark Rowland, Rémi Munos, Mohammad Gheshlaghi Azar, Yunhao Tang, Georg Ostrovski, Anna Harutyunyan, Karl Tuyls, Marc G. Bellemare, and Will Dabney. An analysis of quantile temporal-difference learning. *arXiv*, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Ke Sun, Bei Jiang, and Linglong Kong. How does value distribution in distributional reinforcement learning help optimization? *arXiv*, 2022a.

Ke Sun, Yingnan Zhao, Yi Liu, Bei Jiang, and Linglong Kong. Distributional reinforcement learning via Sinkhorn iterations. *arXiv*, 2022b.

Richard S. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 1984.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

Thibaut Théate, Antoine Wehenkel, Adrien Bolland, Gilles Louppe, and Damien Ernst. Distributional reinforcement learning with unconstrained monotonic neural networks. *arXiv*, 2021.

John N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine learning*, 16(3):185–202, 1994.

John N. Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5): 674–690, 1997.

Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Munchausen reinforcement learning. *Advances in Neural Information Processing Systems*, 2020.

Cédric Villani. *Optimal transport: Old and new*. Springer, 2009.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

Christopher J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge, 1989.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Martha White and Adam White. A greedy approach to adapting the trace parameter for temporal difference learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2016.

Peter R. Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J. Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Kompella, HaoChih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D. Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, 2022.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.

Fan Zhou, Jianing Wang, and Xingdong Feng. Non-crossing quantile regression for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

APPENDICES

We briefly summarise the contents of the appendices for convenience:

- In Section A, we provide further background on QTD, in particular describing the motivation for the form of the algorithm, key theoretical convergence results, and computational considerations.
- In Section B, we provide proofs of all results stated in the main paper.
- In Section C, we provide further details and context on the experimental results reported in the main paper.
- In Section D, we provide further experimental results to complement those presented in the main paper.

A. Further background on quantile temporal-difference learning

As mentioned in the main paper, we encourage readers to consult Rowland et al. (2023) and Bellemare et al. (2023) for detailed background on the QTD algorithm. Here, we present a brief overview of key motivation, intuition, and theoretical results for the algorithm, following the general discussion in Dabney et al. (2018b) and Rowland et al. (2023).

A.1. Motivation

Quantile temporal-difference learning is motivated by the aim of learning certain quantiles of the return distribution at each state of the MDP.

Quantiles. For a probability distribution ν over the real numbers, with corresponding cumulative distribution function (CDF) F which is continuous and strictly increasing, the τ-quantile of ν (for $\tau \in (0, 1)$) is the unique value z such that $F(z) = \tau$. In other words, the τ-quantile of this distribution is the value which a random sample from the distribution has probability exactly τ of being less than. In this way, quantiles can be thought of as an inverse to the CDF, and an alternative description of the probability distribution itself.

Our description above made the assumption that F is continuous and strictly increasing. While this is true for many distributions of interest, such as Gaussians, there are also many distributions for which is is not true, such as distributions of random variables with only finitely many outcomes. The definition of τ-quantile given above cannot apply directly to such CDFs, but the following generalisation applies to all distributions: For any distribution ν over the real numbers with corresponding cumulative distribution function (CDF) F, the *set* of τ-quantiles is defined to be the interval

$$[F^{-1}(\tau), \bar{F}^{-1}(\tau)]\,,$$

where $F^{-1}(\tau) = \inf\{z \in \mathbb{R} : F(z) \geq \tau\}$, and $\bar{F}^{-1}(\tau) = \inf\{z \in \mathbb{R} : F(z) > \tau\}$.

Quantile regression. Quantile regression provides a means of approximately computing the quantiles of a distribution of interest, using samples from this distribution and running stochastic gradient descent on a particular loss function. Just as the mean of a probability distribution ν over the real numbers minimises the squared loss function $\theta \mapsto \mathbb{E}_{Z\sim\nu}[(Z-\theta)^2]$.[†], the τ-quantiles of a distribution ν are precisely the minimisers of the *quantile regression loss*:

$$\theta \mapsto \mathbb{E}_{Z\sim\nu}[|Z-\theta|(\tau \mathbb{1}[Z > \theta] + (1-\tau)\mathbb{1}[Z < \theta])]\,; \tag{7}$$

see e.g. Koenker (2005) or Koenker and Bassett (1978) for further background. Given an estimate θ for the τ-quantile of ν, and a sample Z from θ, a straightforward calculation shows that stochastic gradient descent (SGD) on this loss corresponds to the update

$$\theta \leftarrow \theta + \alpha(\tau - \mathbb{1}[Z < \theta])\,, \tag{8}$$

where α is the step size used in the SGD update. Since the quantile regression loss in Equation (7) is convex in θ, under mild conditions repeated application of the SGD update with a sequence of samples $(Z_i)_{i=1}^{\infty}$ drawn i.i.d. from ν results in the estimate θ converging to the set of τ-quantiles with probability 1 (see e.g. Kushner and Yin, 2003), and this is therefore a sensible algorithm for learning quantiles from a streaming source of samples.

QTD as a combination of quantile regression and bootstrapping. The motivation behind QTD, as proposed by Dabney et al. (2018b), is to use the quantile regression update rule in Equation (8) to learn estimates $(\theta(x, i))_{i=1}^{m}$ of quantiles

[†]To be precise, this characterisation of the mean requires that ν have finite variance, so that the expectation appearing in the definition of the function is finite.

$\tau_i = \frac{2i-1}{2m}$, $i = 1, \dots, m$ of the return distribution at each state $x \in \mathcal{X}$. If we had access to samples $\sum_{t \geq 0} \gamma^t R_t$ drawn from the true distribution of returns at a state x, e.g. from full trajectories of interaction with the environment, then updates of the form given in Equation (8) can be directly applied to each estimate $\theta(x, i)$ separately, leading to an update of the form

$$\theta(x, i) \leftarrow \theta(x, i) + \alpha(\tau_i - \mathbb{1}[\textstyle\sum_{t \geq 0} \gamma^t R_t < \theta(x, i)]) \,.$$

The idea behind QTD is to replace the random return $\sum_{t \geq 0} \gamma^t R_t$ at the state x with an alternative random variable that involves *bootstrapping* (in the sense that the term is used in reinforcement learning, not statistics), in a similar way that classical temporal-difference learning modifies Monte Carlo learning (Sutton and Barto, 2018). Specifically, if the first transition in the trajectory beginning at state x takes us to a new state X', we can replace the portion of the discounted return from this state onwards by instead sampling one of our estimated quantiles of the return distribution at X'. Letting J be a uniformly randomly chosen index in $\{1, \dots, m\}$, this results in an update of the form

$$\theta(x, i) \leftarrow \theta(x, i) + \alpha(\tau_i - \mathbb{1}[R + \gamma\theta(X', J) < \theta(x, i)]) \,.$$

Finally, averaging over the different possible choices for J yields the update

$$\theta(x, i) \leftarrow \theta(x, i) + \alpha\Big(\tau_i - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[R + \gamma\theta(X', j) < \theta(x, i)]\Big) \,,$$

which is precisely the QTD update presented in Algorithm 1. Note that the new return estimator is generally not distributed according to the exact return, and so the QTD update does not directly inherit convergence guarantees from the stochastic gradient descent case described above. Issues of convergence are briefly covered in the section below.

In summary, we have motivated QTD as an algorithm that performs updates similar to those of stochastic gradient descent on the quantile regression, but which uses bootstrapped samples of the random return. This allows for, amongst other things, potential statistical benefits over Monte Carlo learning, as described in the recent work of Cheikhi and Russo (2023), as well as the computation of updates to quantile estimates after each transition in the environment, rather than needing to wait for full trajectories to become available in episodic environments.

A.2. Convergence guarantees

As the theoretical discussion in the main paper centres around convergence points of the QTD update, we review relevant convergence results here at the level of detail required for the paper; for full details, see Rowland et al. (2023).

Rowland et al. (2023) show that under mild conditions, the estimates $(\theta_k(x, i))_{i=1}^m$ produced via k QTD updates from some initial parameters θ_0 are guaranteed to convergence with probability 1 as $k \to \infty$; note that these conditions include the requirement of decaying step sizes, as is often the case with stochastic approximation theory. Unlike classical temporal-difference learning, the estimates of QTD may converge to a *set* of convergence points, rather than a single point; this phenomenon is related to the possibility of non-unique quantiles described above.

Rowland et al. (2023) show that each element of the set of convergence points is the fixed point of a projected distributional Bellman operator $\Pi^\lambda \mathcal{T}^\pi : \mathbb{R}^{\mathcal{X} \times [m]} \to \mathbb{R}^{\mathcal{X} \times [m]}$ that act on the quantile estimates. The precise details of these operators are used in the proof of Proposition B.1 below, so we briefly recount their definitions and key properties. The distributional Bellman operator $\mathcal{T}^\pi$ is typically defined on the space of return-distribution functions $\mathscr{P}(\mathbb{R})^{\mathcal{X}}$ (that is, collections of distributions indexed by state), and outputs the corresponding distributional Bellman targets. Given $\eta \in \mathscr{P}(\mathbb{R})^{\mathcal{X}}$, we have

$$(\mathcal{T}^\pi \eta)(x) = \mathcal{D}_\pi(R + \gamma G(X')) \,,$$

where $(G(y) : y \in \mathcal{X})$ are a collection of random variables with $G(y) \sim \eta(y)$ for all $y \in \mathcal{X}$, (x, R, X') is a random transition beginning at x, and $\mathcal{D}_\pi$ extracts the distribution of the input random variable when the random transition is generated according to π. We can also overload this notation so that $\mathcal{T}^\pi$ takes quantile estimates $((\theta(x, i))_{i=1}^m : x \in \mathcal{X})$ as inputs, first mapping the quantile estimates to the return-distribution function η given by

$$\eta(x) = \sum_{i=1}^{m} \frac{1}{m} \delta_{\theta(x,i)} \,,$$

and then applying the standard definition of the distributional Bellman operator above. With this convention, we have

$$(\mathcal{T}^\pi\theta)(x) = \mathcal{D}_\pi(R + \gamma\theta(X', J))\,,$$

where J is a uniformly random index in $\{1, \dots, m\}$. The projection operator Π^λ, parametrised by $\lambda \in [0,1]^{\mathcal{X}\times[m]}$, then extracts quantiles from the distributions defined by $\mathcal{T}^\pi\theta$, at the levels $\tau_i = \frac{2i-1}{2m}$, for $i = 1, \dots, m$. Mathematically, we have

$$(\Pi^\lambda\mathcal{T}^\pi\theta)(x,i) = (1-\lambda(x,i))F^{-1}_{(\mathcal{T}^\pi\theta)(x)}(\tau_i) + \lambda(x,i)\bar{F}^{-1}_{(\mathcal{T}^\pi\theta)(x)}(\tau_i)\,,$$

where we write F_ν for the CDF associated with distribution ν. Rowland et al. (2023) show that $\Pi^\lambda\mathcal{T}^\pi$ is contractive under a certain Wasserstein metric, and by appealing to the Banach fixed point theorem in an appropriate space of return-distribution functions, show that it has a fixed point. The possibility of multiple convergence points as λ varies is a distinctive property of QTD, and not present in non-distributional algorithms such as classical TD learning. The analysis in this paper relating to these convergence points, such as analysis of fixed-point error, applies to each such point individually, and therefore the question of uniqueness of convergence points is secondary.

A.3. Computational complexity

To complement the description of QTD given above, this section provides further discussion on the computational properties of the algorithm, drawing comparisons with TD.

Space complexity. In addition to the linear scaling with the size of the state space that is common to tabular algorithms, QTD additionally scales linearly with the number of quantiles, m, to be predicted at each state.

Time complexity. An implementation of Algorithm 1 implementing the sum appearing in Line 4 as a for-loop has a time complexity of $O(m^2T)$. One factor of m arises from the fact that there are m quantiles to be updated at each state x_t, and the second factor of m arises from the fact that there are m quantiles at the target state x'_t for which to compute the temporal-difference error.

However, the update for index i in Line 4 of Algorithm 1 depends only on whether $\gamma\theta(x'_t, j)$ is greater than $\theta(x_t, i) - r_t$ for each j. This can be exploited to produce an alternative implementation requiring only $O(Tm\log m)$ time. This implementation first sorts the list $(\gamma\theta(x'_t, j))_{j=1}^m$ in time $O(m\log m)$. Then, for each index i, we can temporarily insert $\theta(x_t, i) - r_t$ into this sorted list at a cost of $O(\log m)$ (via e.g. binary search), and the index of the inserted term reveals exactly how many of the bootstrap terms $(\gamma\theta(x'_t, j))_{j=1}^m$ are less than $\theta(x_t, i) - r_t$, from which the update in Line 4 of Algorithm 1 can be immediately computed. This leads to a per-update time complexity of $O(m\log m)$, improving on the $O(m^2)$ time complexity of the first implementation described above. In contrast to QTD, PQTD requires only linear time to compute each update.

B. Proofs

B.1. Proof of Proposition 4.1

Proposition 4.1. For an MRP with all reward distributions supported on $[R_{\min}, R_{\max}]$, any convergence point θ_m^{QTD} of QTD(m) with corresponding value function estimate $V_m^{\text{QTD}} = (\frac{1}{m}\sum_{i=1}^m \theta_m^{\text{QTD}}(x,i) : x \in \mathcal{X})$ satisfies

$$\|V_m^{\text{QTD}} - V^\pi\|_\infty \leq \frac{R_{\max} - R_{\min}}{2m(1-\gamma)^2}\,.$$

Proof. By Proposition 6.1 of Rowland et al. (2023), we have that for any fixed point $\theta_m^{\text{QTD}} \in \mathbb{R}^{\mathcal{X}\times m}$ of QTD(m), the corresponding return distribution estimate

$$\eta(x) = \frac{1}{m}\sum_{i=1}^m \delta_{\theta_m^{\text{QTD}}(x,i)}$$

satisfies

$$w_1(\eta(x), \eta^\pi(x)) \leq \frac{R_{\max} - R_{\min}}{2m(1-\gamma)^2}\,,$$

for all $x \in \mathcal{X}$, where η^π is the true return-distribution function, and w_1 is the Wasserstein-1 distance. Since the Wasserstein-1 distance between two distributions bounds the distance between their means (see e.g. Villani, 2009), we therefore have

$$\left|\frac{1}{m}\sum_{i=1}^{m}\theta_m^{\text{QTD}}(x,i) - V^\pi(x)\right| \leq \frac{R_{\text{max}} - R_{\text{min}}}{2m(1-\gamma)^2}\,,$$

for all $x \in \mathcal{X}$, as required. □

B.2. Proof of Proposition 4.3

Proposition 4.3. Consider an MRP with all reward distributions having means in $[R_{\text{min}}, R_{\text{max}}]$, and all sub-Gaussian with parameter σ^2, so that $\mathbb{E}_x^\pi[\exp(\lambda(R - \mathbb{E}_x^\pi[R]))] \leq \exp(\lambda^2\sigma^2/2)$, for all $\lambda \in \mathbb{R}$ and $x \in \mathcal{X}$. Then for the value function estimate V_m^{QTD} obtained from any convergence point θ_m^{QTD} of QTD(m) via Equation (4), we have

$$\|V_m^{\text{QTD}} - V^\pi\|_\infty \leq \frac{1}{(1-\gamma)m} \times \left(\frac{R_{\text{max}} - R_{\text{min}} + 2\sigma\sqrt{2\log(2m)}}{2(1-\gamma)} + \frac{\sigma}{\sqrt{2\log(2m)}}\right).$$

We will establish the proof of this result by first deriving the more general Proposition B.1. The proof of this result modifies the approach taken in the proof of Proposition 6.1 from Rowland et al. (2023) in the case of bounded rewards.

Proposition B.1. Consider an MDP with all reward distributions having $\frac{1}{2m}$-quantiles at least $\underline{q}$ and $(1 - \frac{1}{2m})$-quantiles at most $\bar{q}$. Then, for any convergence point θ_m^{QTD} of QTD(m) with corresponding return-distribution function $\eta_m^{\text{QTD}} \in \mathscr{P}(\mathbb{R})^{\mathcal{X}}$, so that

$$\eta_m^{\text{QTD}}(x) = \sum_{i=1}^{m}\frac{1}{m}\delta_{\theta_m^{\text{QTD}}(x,i)}\,,$$

we have

$$w_1(\eta_m^{\text{QTD}}(x), \eta^\pi(x)) \leq \frac{1}{1-\gamma}\left(\frac{\bar{q} - \underline{q}}{2m(1-\gamma)} + \mathbb{E}_x^\pi[(R-\bar{q})\mathbb{1}[R > \bar{q}]] - \mathbb{E}_x^\pi[(R-\underline{q})\mathbb{1}[R < \underline{q}]]\right)$$

for each $x \in \mathcal{X}$.

Proof. As described in Section A.2 (see also Rowland et al., 2023), let $\Pi^\lambda\mathcal{T}^\pi$ be a projected distributional Bellman operator with θ_m^{QTD} as a fixed point, so that $\Pi^\lambda\mathcal{T}^\pi\eta_m^{\text{QTD}} = \eta_m^{\text{QTD}}$. Defining the sup-Wasserstein-1 distance on $\mathscr{P}(\mathbb{R})^{\mathcal{X}}$ (see e.g. Chapter 4, Bellemare et al., 2023) by

$$\overline{w}_1(\eta, \eta) = \max_{x \in \mathcal{X}} w_1(\eta(x), \eta'(x))\,,$$

we have

$$\begin{aligned}\overline{w}_1(\eta_m^{\text{QTD}}, \eta^\pi) \leq& \overline{w}_1(\eta_m^{\text{QTD}}, \mathcal{T}^\pi\eta_m^{\text{QTD}}) + \overline{w}_1(\mathcal{T}^\pi\eta_m^{\text{QTD}}, \eta^\pi)\\ =& \overline{w}_1(\Pi^\lambda\mathcal{T}^\pi\eta_m^{\text{QTD}}, \mathcal{T}^\pi\eta_m^{\text{QTD}}) + \overline{w}_1(\mathcal{T}^\pi\eta_m^{\text{QTD}}, \mathcal{T}^\pi\eta^\pi)\\ =& \overline{w}_1(\Pi^\lambda\mathcal{T}^\pi\eta_m^{\text{QTD}}, \mathcal{T}^\pi\eta_m^{\text{QTD}}) + \gamma\overline{w}_1(\eta_m^{\text{QTD}}, \eta^\pi)\,,\end{aligned}$$

with the first line following from the triangle inequality, the first equality following since $\Pi^\lambda\mathcal{T}^\pi\eta_m^{\text{QTD}} = \eta_m^{\text{QTD}}$ and $\eta^\pi = \mathcal{T}^\pi\eta^\pi$, and the second line following from γ-contractivity of $\mathcal{T}^\pi$ in $\overline{w}_1$ (see e.g. Proposition 4.15, Bellemare et al., 2023). Rearranging, we obtain

$$\overline{w}_1(\eta_m^{\text{QTD}}, \eta^\pi) \leq \frac{1}{1-\gamma}\overline{w}_1(\Pi^\lambda\mathcal{T}^\pi\eta_m^{\text{QTD}}, \mathcal{T}^\pi\eta_m^{\text{QTD}})\,.$$

To bound the right-hand term, we first reason about the support of η_m^{QTD}. Writing $\eta_0(x) = \delta_0$ for all $x \in \mathcal{X}$, and inductively defining $\eta_{k+1} = \Pi^\lambda \mathcal{T}^\pi \eta_k$ for $k \geq 0$, we have $\eta_{k+1} \to \eta_m^{\text{QTD}}$ in $\overline{w}_1$. We now prove by induction that $\eta_k(x)$ is supported on $[\underline{q}\frac{1-\gamma^k}{1-\gamma}, \bar{q}\frac{1-\gamma^k}{1-\gamma}]$ for all $x \in \mathcal{X}$ and $k \geq 0$, from which it follows that $\eta_m^{\text{QTD}}(x)$ is supported on $[\underline{q}\frac{1}{1-\gamma}, \bar{q}\frac{1}{1-\gamma}]$ for all $x \in \mathcal{X}$. The claim is straightforward to see for $k = 0, 1$. For the inductive step, we suppose $\eta_k(x)$ is supported on $[\underline{q}\frac{1-\gamma^k}{1-\gamma}, \bar{q}\frac{1-\gamma^k}{1-\gamma}]$. Now, an instantiation of $(\mathcal{T}^\pi \eta_k)(x)$ is given by $R + \gamma G$, where $(R, X') \sim P^\pi(\cdot|x)$, and $G|R, X' \sim \eta(X')$. Since $G \geq \underline{q}\frac{1-\gamma^k}{1-\gamma}$ almost surely by hypothesis, we have $R + \gamma G \geq R + \gamma \underline{q}\frac{1-\gamma^k}{1-\gamma}$ almost surely. By the definition of $\underline{q}$, we have that the $1/(2m)$-quantile of this distribution is bounded below by $\underline{q} + \gamma \underline{q}\frac{1-\gamma^k}{1-\gamma} = \underline{q}\frac{1-\gamma^{k+1}}{1-\gamma}$, and hence the support of $\Pi^\lambda \mathcal{T}^\pi \eta_k$ is bounded below by $\underline{q}\frac{1-\gamma^{k+1}}{1-\gamma}$. The argument for the upper bound on the support is analogous, and the inductive claim is proven.

Now, define $\Pi_B : \mathscr{P}(\mathbb{R}) \to \mathscr{P}(\mathbb{R})$ as the transformation of distributions that "clips" the support of a distribution to lie on the interval $[\underline{q}\frac{1}{1-\gamma}, \bar{q}\frac{1}{1-\gamma}]$. Formally, if given a probability distribution $\nu \in \mathscr{P}(\mathbb{R})$ we write $\tilde{\nu}$ for its *restriction* to the interval $(\underline{q}\frac{1}{1-\gamma}, \bar{q}\frac{1}{1-\gamma})$, so that $\tilde{\nu}(A) = \nu(A \cap (\underline{q}\frac{1}{1-\gamma}, \bar{q}\frac{1}{1-\gamma}))$, then $\Pi_B \nu$ is given by

$$\Pi_B \nu = \nu((-\infty, \underline{q}\tfrac{1}{1-\gamma}])\delta_{\underline{q}\frac{1}{1-\gamma}} + \tilde{\nu} + \nu([\bar{q}\tfrac{1}{1-\gamma}, \infty))\delta_{\bar{q}\frac{1}{1-\gamma}}\,.$$

Equivalently, Π_B can be defined as the pushforward of ν through the function $f(z) = \max(\min(z, \bar{q}\frac{1}{1-\gamma}), \underline{q}\frac{1}{1-\gamma})$, i.e. $\Pi_B \nu = f_\# \nu$. We then use the triangle inequality to write

$$\overline{w}_1(\Pi^\lambda \mathcal{T}^\pi \eta_m^{\text{QTD}}, \mathcal{T}^\pi \eta_m^{\text{QTD}}) \leq \overline{w}_1(\Pi^\lambda \mathcal{T}^\pi \eta_m^{\text{QTD}}, \Pi_B \mathcal{T}^\pi \eta_m^{\text{QTD}}) + \overline{w}_1(\Pi_B \mathcal{T}^\pi \eta_m^{\text{QTD}}, \mathcal{T}^\pi \eta_m^{\text{QTD}})\,.$$

Now, each distribution $(\mathcal{T}^\pi \eta_m^{\text{QTD}})(x)$ has at most $1/(2m)$ mass in the region $(-\infty, \underline{q}/(1-\gamma)]$, and similarly for the region $[\bar{q}/(1-\gamma), \infty)$, so $\Pi^\lambda \Pi_B \mathcal{T}^\pi \eta_m^{\text{QTD}}(x) = \Pi^\lambda \mathcal{T}^\pi \eta_m^{\text{QTD}}(x)$, and so we obtain the bound

$$\overline{w}_1(\Pi^\lambda \mathcal{T}^\pi \eta_m^{\text{QTD}}, \Pi_B \mathcal{T}^\pi \eta_m^{\text{QTD}}) \leq \frac{\bar{q} - \underline{q}}{2m(1-\gamma)}$$

from the argument given by Rowland et al. (Proposition 6.1, 2023). To quantify the contribution of the clipping to the second Wasserstein distance term, we first bound the transport cost for the upper clipping as

$$\begin{aligned}
&\mathbb{E}[(R + \gamma\theta(X', J) - \bar{q}/(1-\gamma))\mathbb{1}[R + \gamma\theta(X', J) > \bar{q}/(1-\gamma)]] \\
\leq &\mathbb{E}[(R + \gamma\bar{q}/(1-\gamma) - \bar{q}/(1-\gamma))\mathbb{1}[R + \gamma\theta(X', J) > \bar{q}/(1-\gamma)]] \\
\leq &\mathbb{E}[(R + \gamma\bar{q}/(1-\gamma) - \bar{q}/(1-\gamma))\mathbb{1}[R + \gamma\bar{q}/(1-\gamma) > \bar{q}/(1-\gamma)]] \\
= &\mathbb{E}[(R - \bar{q})\mathbb{1}[R > \bar{q}]]\,.
\end{aligned}$$

The bound for the transport cost associated with the lower clipping is derived analogously, and the result follows. □

Corollary B.2. Under the same conditions as Proposition B.1, we have that V_m^{QTD}, the value prediction obtained from the fixed point θ_m^{QTD}, satisfies

$$|V^{\text{QTD}}(x) - V^\pi(x)| \leq \frac{1}{1-\gamma}\left(\frac{\bar{q} - \underline{q}}{2m(1-\gamma)} + \mathbb{E}_x^\pi[(R - \bar{q})\mathbb{1}[R > \bar{q}]] - \mathbb{E}_x^\pi[(R - \underline{q})\mathbb{1}[R < \underline{q}]]\right)$$

for each $x \in \mathcal{X}$.

Proof. As in the proof of Proposition 4.1, we use the fact that the Wasserstein-1 distance between two distributions bounds the difference in mean of the distributions, and the result now follows from Proposition B.1. □

Proposition B.1 is stated abstractly, and to obtain concrete bounds for reward distributions of interest, it is necessary to obtain bounds on the quantiles and conditional expectations that feature in the result. We give a concrete result below for sub-Gaussian distributions, which proves Proposition 4.3.

Proposition B.3. Consider an MDP with all reward distributions having means in $[R_{\min}, R_{\max}]$, and all sub-Gaussian with parameter σ^2, so that $\mathbb{E}_x^\pi[\exp(\lambda(R - \mathbb{E}_x^\pi[R]))] \leq \exp(\lambda^2\sigma^2/2)$, for all $x \in \mathcal{X}$. Then

$$\overline{w}_1(\eta_m^{\text{QTD}}, \eta^\pi) \leq \frac{1}{(1-\gamma)m}\left(\frac{R_{\max} - R_{\min} + 2\sigma\sqrt{2\log(2m)}}{2(1-\gamma)} + \frac{\sigma}{\sqrt{2\log(2m)}}\right),$$

and

$$\|\hat{V} - V^\pi\|_\infty \leq \frac{1}{(1-\gamma)m}\left(\frac{R_{\max} - R_{\min} + 2\sigma\sqrt{2\log(2m)}}{2(1-\gamma)} + \frac{\sigma}{\sqrt{2\log(2m)}}\right).$$

Proof. By a standard conversion of the sub-Gaussian moment-generating function condition into a concentration inequality, we obtain bounds for $\bar{q}$ and $\underline{q}$ in Proposition B.1 of the form

$$\bar{q} = R_{\max} + \sigma\sqrt{2\log(2m)}\,, \quad \underline{q} = R_{\min} - \sigma\sqrt{2\log(2m)}\,.$$

Next, we can also compute a bound on the expectations appearing in Proposition B.1 as follows.

$$\begin{aligned}
\mathbb{E}[(R-\bar{q})\mathbb{1}[R > \bar{q}]] &\leq \int_{\sigma\sqrt{2\log(2m)}}^{\infty} \exp(-t^2/(2\sigma^2))\mathrm{d}t \\
&\leq \int_{\sigma\sqrt{2\log(2m)}}^{\infty} \frac{t}{\sigma\sqrt{2\log(2m)}} \exp(-t^2/(2\sigma^2))\mathrm{d}t \\
&= \frac{\sigma}{\sqrt{2\log(2m)}} \exp(-(\sigma\sqrt{2\log(2m)})^2/(2\sigma^2)) \\
&= \frac{\sigma}{2m\sqrt{2\log(2m)}}\,,
\end{aligned}$$

where the first inequality follows from the bounds on the quantiles and CDF tails for the sub-Gaussian reward distribution, and the second inequality is a standard trick for bounding tails of a Gaussian CDF. The same bound for the second expectation is derived analogously, and we obtain the required statement by substituting into the expression in Proposition B.1. □

B.3. Proof of Proposition 4.4

Proposition 4.4. Consider an MDP with all reward distributions having finite mean. Then for the value function estimate V_m^{QTD} obtained from any convergence point θ_m^{QTD} of QTD(m) via Equation (4), we have $\|V_m^{\text{QTD}} - V^\pi\|_\infty \to 0$ as $m \to \infty$.

To prove this result, we first require the following lemma, which controls how fast the tails of a distribution's CDF can decay if the distribution has finite mean.

Lemma B.4. If F is the CDF of a probability distribution over the real numbers with finite mean, then we must have

$$F^{-1}(1 - \tfrac{1}{2m}) = o(m)\,, \quad F^{-1}(\tfrac{1}{2m}) = o(m)\,.$$

Proof. We will prove the claim for $F^{-1}(1 - \frac{1}{2m})$; the proof for $F^{-1}(\frac{1}{2m})$ is analogous. First, note that for a random variable Z with CDF F, we have

$$\mathbb{E}[Z\,\mathbb{1}[Z > 0]] = \int_{F(0)}^{1} F^{-1}(\tau)\,\mathrm{d}\tau\,.$$

Suppose for a contradiction the growth bound on $F^{-1}(1 - \frac{1}{2m})$ does not hold. Then there exists $c > 0$ such that for infinitely many m, $F^{-1}(1 - \frac{1}{2m}) > cm$. We will use this fact to lower-bound the value of the integral above. Note that for each such m, the rectangle $C_m = [1 - \frac{1}{2m}, 1] \times [0, F^{-1}(1 - \frac{1}{2m})]$ lies between the curve to be integrated and the x-axis, and has area at least $c/2$. Now we pick a subsequence of these integers $m_1, m_2, \ldots$, with the property that the area of $C_{m_{k+1}} \cap C_{m_k}$ is less than $c/4$. Concretely, this can be achieved by taking $m_{k+1} > 2c^{-1}F^{-1}(1 - \frac{1}{2m_k})$. We then obtain the bound

$$\int_{F(0)}^{1} F^{-1}(\tau)\,\mathrm{d}\tau \geq \text{Area}(C_1) + \sum_{k=2}^{\infty} \text{Area}(C_{m_k} \setminus C_{m_{k-1}}) \geq c/2 + c/4 + c/4 + \cdots = \infty\,.$$

However, this contradicts our initial assumption that the distribution has finite mean, and so we conclude that we must have $F^{-1}(1-\frac{1}{2m}) = o(m)$, as required. □

Proof of Proposition 4.4. By Corollary B.2 and Lemma B.4, for m sufficiently large we have

$$|V_m^{\mathrm{QTD}}(x) - V^\pi(x)| \leq \frac{1}{1-\gamma}\Big(\frac{o(m)}{2m(1-\gamma)} + \mathbb{E}_x^\pi[(R - F^{-1}(1-\tfrac{1}{2m}))\mathbb{1}[R > F^{-1}(1-\tfrac{1}{2m})]] - \mathbb{E}_x^\pi[(R - F^{-1}(\tfrac{1}{2m}))\mathbb{1}[R < F^{-1}(\tfrac{1}{2m})]]\Big) \to 0\,,$$

as required, where F above is the CDF of the reward distribution at state x. □

C. Further experimental details

We provide full details for the experiments reported in the main paper.

Environment details.

- Dense stochastic transition structure. We generate a transition matrix for a 20 state MDP by sampling each row of the transition matrix independently from a Dirichlet$(1, \ldots, 1)$ distribution.
- Sparse transition structure. We generate a transition matrix for a 20 state MDP according to the Garnet protocol (Archibald et al., 1995). Specifically, for each state, we independently sample 6 states (without replacement) to have non-zero transition probability, and allocate the transition probability uniformly across these states.
- Deterministic transition structure. We use a deterministic cycle transition structure over 10 states.

Reward distributions.

- Deterministic distributions. At the time of generating the environment, we sample the reward at each state independently from a standard normal $N(0, 1)$ distribution.
- Gaussian distributions. Mean rewards are sampled as above, and the distributions themselves are Gaussian with standard deviation 1.
- Exponential distributions. Mean rewards are sampled as above, and the distributions themselves are shifted Exponential(1) distributions with the specified means.
- t-distributions. Mean rewards are sampled as above, and the distributions themselves are shifted t_2-distributions with the specified means.

Hyperparameters. In all experiments, we use a default discount factor of $\gamma = 0.9$. For both TD and QTD methods, all predictions are initialised to 0.

Mean-squared error measurement. Each configuration was run 1,000 times, and the reported results are the mean-squared errors averaged over these 1,000 runs; error bars in the plots correspond to plus/minus two times the empirical standard error.

Learning rates. For TD, 40 learning rates are swept over the range $[5 \times 10^{-4}, 1]$, equally spaced in log-space. For QTD, 40 learning rates are swept over the range $[5 \times 10^{-3}, 10]$, equally spaced in log-space.

D. Further experimental results

Here, we report further comparisons to complement the results in the main paper.

D.1. Optimal learning rate selection for Figure 3

To aid interpretation of Figure 3, we display in Figure 9 the optimal learning rates selected by TD and QTD for each number of updates. Here, the confidence bands indicate the range of learning rates for which the lower end of the MSE confidence region was smaller than the upper end of the confidence region for the actual chosen optimal learning rate.

D.2. Further results for t_2-distributed rewards

To complement Figure 4, we plot mean-squared error against learning rates for both TD and QTD run with 1,000 updates in Figure 10, in particular providing an indication of the typical magnitude of MSE that is attained by each method.

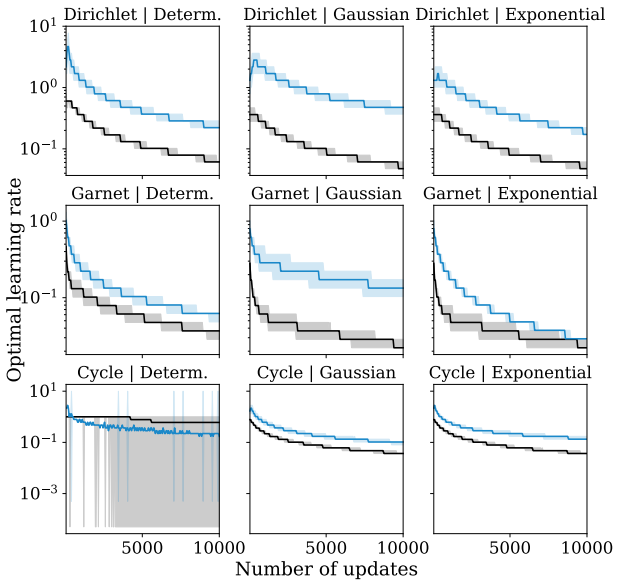

Figure 9. Optimal learning rate, as a function of number of updates, for both TD and QTD, for the results displayed in Figure 3.

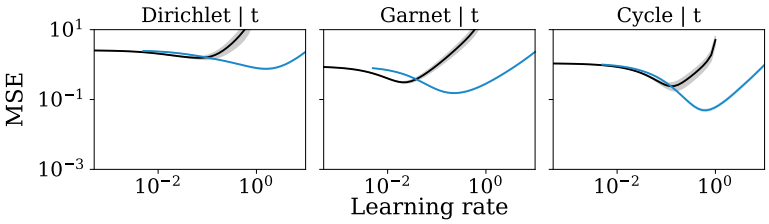

Figure 10. Mean-square error against learning rate for both TD and QTD run for 1,000 updates in environments with t_2-distributed rewards.

D.3. Varying numbers of quantiles

We present results for the relative improvement of QTD(1) and QTD(16) over TD on the main suite of environments in Figures 11 and 12, respectively. Unlike QTD(128), which is superior to TD on all stochastic environments in the suite, QTD(1) is outperformed by TD on several stochastic environments, showing that performance can be degraded when not using a sufficient number of quantiles. Interestingly, there are also environments in which the performance of QTD(1) is not degraded relative to QTD(128). We note in particular that the performance of QTD(1) on the cycle Gaussian environment in fact improves over that of QTD(128). In fact, in environments with deterministic transitions and certain symmetric reward distributions, there is no fixed-point error in the value predictions of QTD(1); this is intuitively driven by the agreement of the median and the mean of the distributional Bellman targets in this environment, in contrast to e.g. the example in Figure 5. We note also the oscillations that appear in certain environments in Figure 11; these artefacts are due to the discrete grid of learning rates that are swept over. Unlike QTD(1), QTD(16) dominates TD in the stochastic environments within the suite, and the performance is comparable with that of QTD(128), indicating the diminishing returns of increasing the number of quantiles to be estimated beyond a certain range.

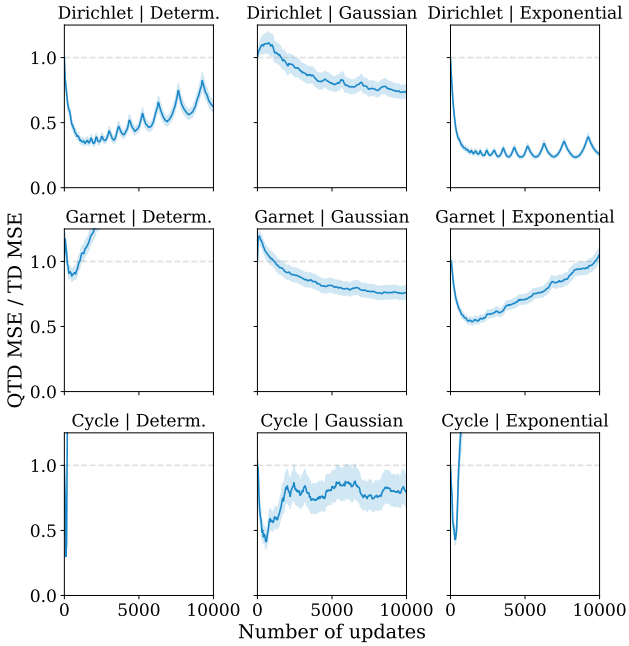

Figure 11. Relative improvement of QTD(1) over TD in mean-squared error against number of updates.

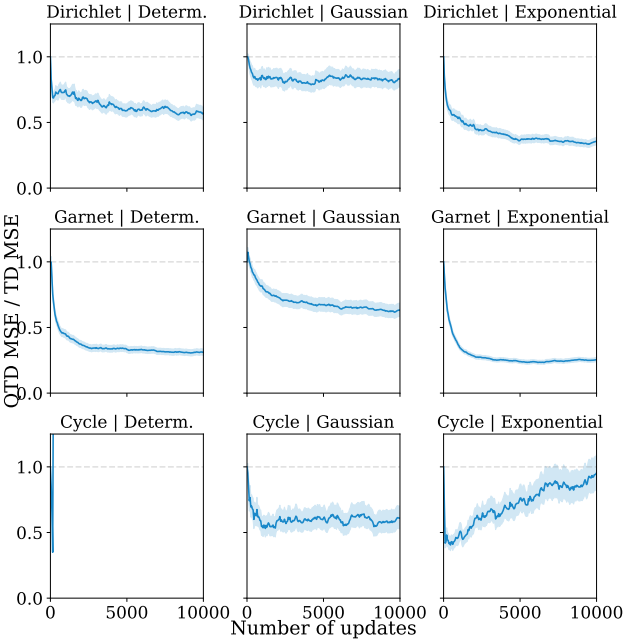

Figure 12. Relative improvement of QTD(16) over TD in mean-squared error against number of updates.